Complémentarité de DNSsec et d'IPsec

Gilles Guette, Olivier Courtay, Bernard Cousin

Complémentarité de DNSsec et d'IPsec

Gilles GUETTE
IRISA/INRIA, Campus de beaulieu, 35042 Rennes Cedex, France
gilles.guette@irisa.fr

Olivier COURTAY
ENST-Bretagne, 2 rue de la Châtaigneraie, 35512 Cesson Sévigné Cedex, France
olivier.courtay@enst-bretagne.fr

Bernard COUSIN
IRISA/Université de Rennes I, Campus de beaulieu, 35042 Rennes Cedex, France
bernard.cousin@irisa.fr

Résumé

Pour sécuriser le protocole DNS (Domain Name System) l'IETF (Internet Engineering Task Force) a défini DNSsec, le protocole DNS sécurisé. DNSsec est une extension du protocole DNS. Il conserve la même structure arborescente et les mêmes entités : résolveurs, serveurs de noms et cache/forwarder. Il ajoute des enregistrements spécifiques pour la sécurisation du DNS.
Nous allons voir que malgré la mise en œuvre de services sécurisés au sein de DNSsec, il importe d'utiliser conjointement IPsec, le protocole IP sécurisé pour atteindre un niveau de sécurité suffisant pour DNSsec mais aussi pour IPsec.

Mots clefs

DNSsec, IPsec, *Opportunistic Encryption*, intégrité, authentification.

1 Introduction

L'infrastructure DNS (*Domain Name System*) [Moc87a, Moc87b, AL02] est une ressource critique de l'utilisation d'Internet. Le *Domain Name System* est une base de données distribuée dont le but est entre autres, de fournir un service de traduction appelé résolution de noms. L'architecture du DNS repose sur une structure arborescente (figure 1), l'arbre DNS est composé de plusieurs domaines : un sous-arbre partant d'un nœud jusqu'aux feuilles. Chaque domaine est composé de zones : chaque nœud de l'arbre. Chaque zone stocke ses informations locales et est sous la responsabilité de l'administrateur de la zone.

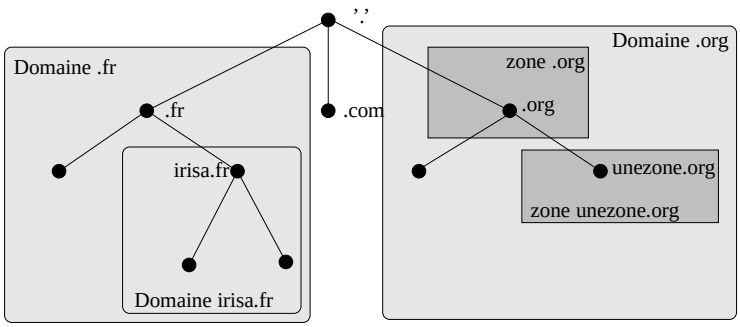


FIG. 1 – *La structure arborescente du DNS.*

Le DNS est une ressource critique et nécessaire au bon fonctionnement des applications Internet, c'est pourquoi l'IETF (*Internet Engineering Task Force*) a défini le protocole DNSsec [Eas99a, Gun03] qui est l'évolution du protocole DNS en une version sécurisée.
DNSsec s'appuie sur une cryptographie à clé publique pour fournir deux services majeurs de sécurité : l'intégrité et l'authentification. Ces deux services permettent de protéger les données et les transactions DNS.

Conjointement, il est possible d'utiliser IPsec [TDG98] pour protéger les communications entre les différents serveurs DNS, notamment les messages de mise à jour dynamique [Wel00] et les messages de transfert de zone. IPsec est la version sécurisée du protocole IP fournissant les services nécessaires à la sécurisation de ce protocole.
Nous présentons dans une première partie le protocole DNSsec, puis dans une seconde partie nous décrivons le protocole IPsec et enfin nous explicitons les interactions possibles entre IPsec et DNSsec, afin de sécuriser le DNS le plus efficacement possible, ainsi que leur utilisation pour la mise en place d'autres services tel que l'*Opportunistic Encryption*.

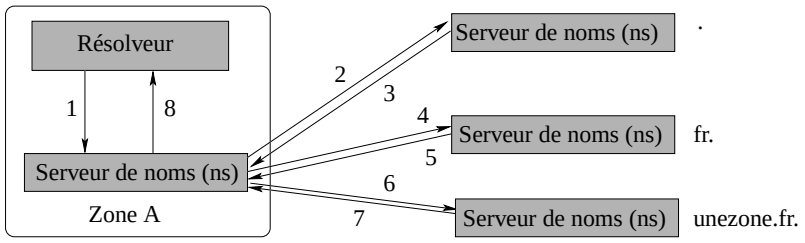
2 DNS et DNSsec

2.1 Le Domain Name System

Le DNS est une base d'informations distribuée permettant d'effectuer la traduction de noms, c'est-à-dire passer d'un nom de machine à une adresse IP et inversement. Deux entités du DNS servent à la résolution de noms : le *serveur de noms* qui maintient les informations sur la zone et le *résolveur* qui extrait l'information du serveur de noms pour répondre à la requête d'une application cliente. Le résolveur se situe sur la même machine que l'application.

2.2 Le serveur de noms

Le serveur de noms a la charge de stocker les informations d'une zone dans le fichier de zone. Sa fonction essentielle est de répondre aux requêtes en utilisant les données de son fichier de zone. Pour répondre aux requêtes, il existe deux possibilités, l'une itérative (par défaut) et l'autre récursive. La méthode de résolution est déterminée par deux choses, le bit RD (*Recursion Desired*) de l'en-tête et la configuration du serveur recevant la requête. Si la méthode de résolution est itérative (le bit RD n'est pas positionné), le serveur de noms retourne la meilleure réponse qu'il possède, c'est-à-dire la réponse finale ou alors le nom et l'adresse d'un autre serveur de noms à contacter et c'est le demandeur qui continue la résolution de nom. Si la méthode est récursive (le bit RD est positionné et le serveur interrogé sait effectuer une résolution récursive), le serveur de noms fait suivre lui-même la requête au serveur le plus à même de répondre, obtient la réponse et retourne la réponse au demandeur.
Lorsqu'une requête parvient au serveur de noms (par l'intermédiaire d'un résolveur ou d'un autre serveur de noms), deux cas peuvent se présenter. Soit le serveur de noms interrogé possède la réponse et l'envoie, soit il ne la possède pas mais il connaît le nom ou l'adresse d'un autre serveur de noms à contacter. Il envoie alors les informations sur cet autre serveur de noms en réponse. C'est ici qu'intervient la structure arborescente. Le serveur de noms connaît toutes les informations relatives à sa zone ainsi que les noms et adresses des serveurs de noms de ses zones filles. On peut voir sur la figure 2 que pour le résolveur le processus de résolution est récursif tandis que du point de vue du serveur le processus de résolution est itératif.



1 : unemachine.unezone.fr ?
2 : unemachine.unezone.fr ?
3 : ns.fr
4 : unemachine.unezone.fr ?
5 : ns.unezone.fr
6 : unemachine.unezone.fr ?
7 : unemachine.unezone.fr = @IP
8 : unemachine.unezone.fr = @IP

FIG. 2 – *Les processus de résolution.*

Deux critères sont essentiels pour un serveur de noms : la rapidité de réponse et la fiabilité du service. Pour assurer

la rapidité de réponse les serveurs de noms et/ou les résolveurs possèdent généralement un cache dans lequel ils mémorisent les enregistrements auxquels ils ont eu accès récemment. Afin d'assurer la fiabilité du service DNS, la politique d'implantation des serveurs requiert généralement que toutes les zones soient supportées par plusieurs serveurs de noms. Le ou les serveurs secondaires assurent la redondance du serveur primaire.

2.3 Le résolveur

Le résolveur se situe entre l'application qui demande la résolution du nom et le serveur de noms. Il est situé sur la même machine que l'application. C'est lui qui reçoit la requête de l'application, interroge le serveur de noms et renvoie la réponse à l'application. Il est généralement composé d'un ensemble de routines il délègue la totalité du travail aux serveurs de noms. Ce type de résolveur est appelé résolveur basique ou *stub resolver*. Certains résolveurs peuvent disposer de leur propre cache pour stocker les données auxquelles ils ont déjà accédées car l'un des critères important d'un résolveur est de minimiser voire d'éliminer les délais du réseau et de résolution des requêtes. Il y parvient en répondant immédiatement lorsqu'il le peut grâce aux données stockées dans son cache. Les données en cache restent valides un certains temps, déterminé par le serveur de noms.
Il existe un second type de résolveurs, les résolveurs complets ou *Full Resolver*. Un résolveur complet possède les ressources nécessaires et tout le matériel cryptographique pour effectuer les vérifications DNSsec lui-même.

2.4 DNSsec

Le DNS sécurisé, DNSsec, s'appuie sur une cryptographie à clé publique et sur des signatures numériques. Il a pour objectif de garantir l'intégrité des enregistrements DNS et leur authentification. Pour utiliser ces mécanismes et générer les signatures numériques, chaque zone possède au moins une paire de clés publique/privée appelées clés de zone. Ces clés servent à signer les enregistrements contenus dans le fichier de zone. En effet, étant un service public il n'y a nul besoin de crypter les données du DNS.
Chaque serveur doit être capable de supporter l'existence de deux paires de clés simultanément pour assurer le recouvrement temporel lors du changement d'une paire de clés arrivant à expiration.
Pour stocker les clés, les signatures et des données nécessaires au DNSsec, quatre types d'enregistrement ont été ajoutés au DNS. Il s'agit des enregistrements de types KEY, SIG, NXT [Eas99a] et DS [Gun03]. Le rôle de l'enregistrement KEY est de rendre les clés publiques de zones disponibles. Il est chargé de contenir une clé publique de zone, les informations concernant les différentes utilisations possibles de cette clé, ainsi que les protocoles et les algorithmes avec lesquels elle peut être utilisée. La clé privée servant à générer les signatures des enregistrements est conservée dans un endroit sûr. Les signatures seront stockées dans un enregistrement spécifique au DNSsec, l'enregistrement de type SIG.
L'enregistrement de type NXT a été créé pour envoyer une réponse sécurisée à une question portant sur un nom ou des enregistrements qui n'existent pas. Avec le protocole DNS, si un résolveur demande l'adresse d'une machine qui n'existe pas, il recevra en réponse un message contenant un code d'erreur dans l'en-tête du message DNS, sans signature vérifiable. Pour éviter cette réponse non signée et donc non vérifiable, DNSsec possède un nouveau type d'enregistrement, l'enregistrement NXT. Cet enregistrement contient un vecteur de bits indiquant tous les types d'enregistrements existant pour le nom associé à l'enregistrement NXT, ainsi que le prochain nom se trouvant dans le fichier de zone selon l'ordre établi (ordre lexicographique sur les étiquettes et de l'étiquette la plus à droite vers celle la plus à gauche).
Un enregistrement NXT possède son enregistrement SIG associé, il est alors impossible de modifier l'enregistrement NXT sans en modifier la signature. Il est ainsi possible de déduire de manière sécurisée qu'un nom n'existe pas, s'il se situe entre le nom associé à l'enregistrement NXT et le nom contenu dans l'enregistrement NXT, ou qu'un type d'enregistrement n'existe pas, si le bit associé à ce type est à 0 dans le vecteur de bits de l'enregistrement NXT reçu en réponse.

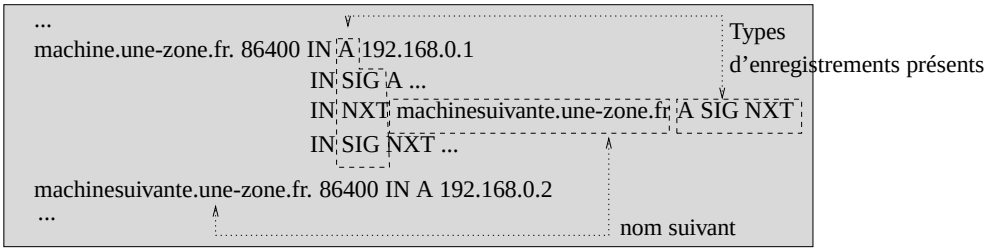


FIG. 3 – *L'enregistrement NXT.*

Dans l'extrait de fichier de zone représenté sur la figure 3, il est possible de déduire que le nom *machineperso.une-zone.fr* n'existe pas car il se situe entre *machine.une-zone.fr* et *machinesuivante.une-zone.fr*. Et que par exemple, il n'existe pas d'enregistrement KEY pour le nom *machine.une-zone.fr*.
Le quatrième type d'enregistrement créé est l'enregistrement DS. Il est utilisé pour rendre le lien entre une zone parente et sa zone fille vérifiable et ainsi créer un chaîne de confiance [Gie01]. Le terme *chaîne de confiance* désigne le lien vérifiable au sens cryptographique que peut construire un serveur de noms ou un résolveur à partir d'un point d'entrée sécurisé (la racine de l'arbre par exemple) jusqu'à la ressource recherchée. L'enregistrement DS est stocké dans la zone parente. Il contient entre autres l'identifiant de la clé publique de la zone fille et un hachage de diverses informations (nom de la zone fille, clé, etc.). Pour obtenir l'enregistrement DS on procède de la manière suivante : l'administrateur de la zone fille signe l'ensemble de ses clés avec sa ou ses clés privées et envoie le KEY RRset de la zone fille (ensemble des clés publiques) à l'administrateur de la zone parente. Ce dernier crée ou supprime les enregistrements DS correspondant qui sont conservés dans la zone parente. Il n'y a qu'un seul message envoyé pour générer l'enregistrement DS et un nouveau message sera envoyé pour modifier l'enregistrement DS uniquement lorsque l'on changera une clé de zone de la zone fille, qui a signé son enregistrement KEY.

2.5 Transaction

On peut distinguer plusieurs types de communications au sein du DNS :
– les requêtes/réponses entre un résolveur et un serveur ou entre deux serveurs,
– les transferts de zone entre un serveur primaire et un serveur secondaire,
– les messages de mise à jour à destination d'un serveur primaire.
Grâce aux mécanismes présentés précédemment, DNSsec est capable de garantir l'intégrité des données stockées sur les différents serveurs de noms ainsi que l'authentification de la source de ces données. Néammoins, les signatures numériques protègent les enregistrements auxquels elles sont associées mais ne fournissent aucune garantie sur l'intégrité de l'intégralité des messages envoyés. L'en-tête du message n'étant pas signé, il peut donc être modifié, notamment les champs précisant le nombre d'enregistrements dans les différentes sections du message. Il est ainsi possible d'insérer des enregistrements illégaux (cependant leur signature et leur date de validité doivent être correctes, il s'agit d'un rejeu). Pour éviter de telles manipulations, il existe deux mécanismes de signature de l'intégralité des messages DNS. Il s'agit de TSIG [VGEW00] et de SIG(0) [Eas00].
TSIG ajoute un enregistrement supplémentaire qui contient la signature de l'intégralité du message et des variables spécifiques à TSIG. Ce mécanisme garantit l'authentification et l'intégrité du message entier et utilise des mécanismes de cryptographie à clé symétrique. L'utilisation d'algorithmes à clé symétrique a l'avantage de rendre le traitement cryptographique plus rapide mais il y a tout de même un inconvénient : ce mécanisme ne peut pas passer à l'échelle à cause du nombre de clés à gérer (une clé pour chaque couple de machines). TSIG peut être utilisé entre un serveur primaire et ses serveurs secondaires pour sécuriser le transfert de zone, le nombre de serveurs de noms restant raisonnable pour la plupart des zones. Il est néammoins nécessaire de pouvoir échanger de manière sécurisée un secret partagé.
Un autre mécanisme existe pour signer les transactions : SIG(0) [Eas00]. SIG(0) fonctionne exactement comme l'enregistrement SIG, c'est-à-dire qu'il contient une signature. Le 0 indique qu'il ne s'applique pas à un enregistrement particulier mais à tout le message. Il contient donc la signature de tout le message y compris l'en-tête et tout comme les enregistrements de type SIG il est créé grâce à la clé privée de la zone. Une attention particulière doit être portée sur la protection de la clé privée qui doit alors être gardée disponible en permanence sur le serveur de noms pour permettre la signature des messages.

Si TSIG ou SIG(0) sont utilisés, un tiers ne peut modifier ou ajouter des informations dans le message sans en modifier la signature. Si la signature n'est pas correcte le message DNS est rejeté. Malheureusement ces mécanismes ne sont pas utilisables systématiquement. En effet, l'utilisation de la cryptographie pour signer chaque message puis vérifier les signatures prend du temps et impliquerait une forte dégradation des temps de réponse et des performances du DNS. C'est pourquoi nous nous sommes tournés vers IPsec qui fournit naturellement les services de sécurité nécessaire.

3 Le protocole IPsec

Le protocole IP [Pos81] est le protocole de communication à la base de l'Internet. IP n'est pas un protocole fiable c'est-à-dire qu'il n'assure pas l'ordre d'arrivée des paquets transmis et ne s'assure pas non plus que ces paquets sont bien arrivés à destination. IP est basé sur le principe de *best effort*, c'est-à-dire que chaque équipement d'interconnexion fait de son mieux pour faire suivre les paquets. Il va sans dire qu'IP n'intègre aucun mécanisme ni service de sécurité, ainsi les paquets IP et leur contenu circulent en clair sur le réseau et peuvent être la cible de multiples attaques : écoute, usurpation d'adresse, modification, etc. Il est apparu nécessaire de définir des mécanismes de sécurité à inclure dans le protocole IP. Cela s'est traduit par la définition du protocole IPsec [TDG98] qui est inclus dans le protocole IPv6 mais optionnel dans le protocole IPv4.

3.1 Principe et fonctionnement

IPsec fournit quatre services de sécurité [Sch96] :
– *L'authentification* : le destinataire d'un message doit pouvoir s'assurer de son origine. Un individu malveillant ne doit pas être capable de se faire passer pour la source légitime.
– *L'intégrité* : le destinataire d'un message doit pouvoir vérifier que celui-ci n'a pas été modifié durant son acheminement. Un utilisateur malveillant doit être incapable de faire passer un message modifié pour légitime.
– *La confidentialité* : un utilisateur malveillant interceptant un paquet ne doit pas être en mesure d'interpréter les données contenues dans ce paquet.
– *La non-répudiation* : l'expéditeur d'un message ne doit pas pouvoir, par la suite, nier avoir envoyé un message.
Pour cela IPsec possède trois méthodes de fonctionnement : AH (*Authentication Header*) [KA98a], ESP (*Encapsulation Security Payload*) [KA98b] et IPcomp (*IP compression*). AH permet l'authentification et garantit l'intégrité d'un paquet IP, ESP assure en plus la confidentialité. IPcomp permet de compresser les données qui transitent.
Détaillons les différentes entités en place pour pouvoir initier une communication IPsec.

La *Security Policy Database* (SPD). Pour pouvoir initier une communication IPsec, il faut que cette communication respecte la politique de sécurité mise en place pour la machine communicante. La *Security Policy Database* est la base de données qui contient toutes les politiques de sécurité connues de la machine. Elle est configurée par l'entité en charge de l'administration de ces politiques.

La *Security Association Database* (SADB). Lorsqu'une machine utilise IPsec, elle n'est pas limitée à une seule connexion ou à une seule communication et chacune de ces communications possède sa propre politique de sécurité négociée lors de l'établissement de la connexion. La SADB garde un contexte de chaque connexion afin de pouvoir conserver la politique de sécurité associée à chaque communication. Un contexte est appelé une SA (*Security Association*), les SA sont conservés dans la SADB et contiennent :
– l'identifiant de connexion ou SPI (*Security Parameter Index*),
– l'adresse de destination des paquets,
– le mécanisme IPsec utilisé (AH ou ESP),
– les algorithmes utilisés pour ces mécanismes.
Si AH et ESP sont utilisés conjointement, plusieurs SA seront liées à cette connexion (une pour AH et une pour ESP). Il en est de même si la connexion est bidirectionnelle.

3.2 Les modes IPsec

Il existe deux modes différents dans IPsec, le mode transport et le mode tunnel. Ces deux modes diffèrent par la construction des paquets IP. Dans le mode transport, seules les données contenues dans la couche transport sont protégées par IPsec, l'en-tête IP reste inchangé. Dans le mode tunnel, tout le paquet IP est protégé. Pour cela il est considéré comme un simple message et un nouvel en-tête IP est créé.
Le mode tunnel possède comme grand intérêt de pouvoir créer des tunnels sécurisés. Deux machines ou passerelles

de deux réseaux voulant sécuriser leurs communications qui passent par une zone non protégée, vont créer un tunnel IPsec entre ces deux passerelles. Toute communication d'une machine d'un réseau vers l'autre sera encapsulée dans un nouvel en-tête IP. Une personne écoutant la communication ne verrait que des paquets allant d'une passerelle à l'autre sans pouvoir déchiffrer le contenu de ces paquets. Ce mode correspond à l'utilisation d'un réseau privé virtuel ou VPN (*Virtual Private Network*)

3.3 La gestion des clés dans IPsec

Les mécanismes AH et ESP utilisent la cryptographie, il faut donc pouvoir échanger les clés nécessaires à l'utilisation des mécanismes cryptographique. C'est le rôle du protocole IKE (*Internet Key Exchange*) [HC98] qui effectue la négociation des SA.
Pour gérer les SA et les clés de chiffrement, l'IETF (*Internet Engineering Task Force*) distingue :
– un protocole de gestion des SA, ISAKMP (*Internet Security Association and Key Management Protocol*) [MSST98],
– un protocole d'échange de clés de session comme SKEME [Kra96] ou Oakley [Orm98],
– des infrastructures à clés publiques.
Le principe de fonctionnement d'IKE est qu'un paquet envoyé par une machine est confronté à la SPD. Si ce paquet dépend d'IPsec et qu'aucune SA ne lui est associée, le système utilise IKE pour mettre en place la SA correspondante. ISAKMP n'est pas spécifique à IPsec, il existe donc un domaine d'interprétation (DOI) qui permet de définir les conventions d'utilisation d'ISAKMP dans IPsec. ISAKMP se décompose en deux phases :

1. La première phase permet de vérifier l'identité des entités en présence. Les machines décident des algorithmes de cryptographie utilisés pour les futures négociations ISAKMP. À la fin de cette phase, chaque entité doit disposer d'une clé de chiffrement, d'une clé d'authentification et d'un secret partagé.
2. La seconde phase permet de négocier les attributs plus spécifiques à IPsec (utilisation d'AH ou d'ESP par exemple). Ces échanges sont chiffrés et authentifiés grâce aux éléments décidés lors de le première phase.

La première phase fait appel à de la cryptographie asymétrique qui est lente. Elle n'est utilisée qu'une seule fois pour définir les paramètres qui vont permettre de sécuriser les échanges de la seconde phase. Lors de la première phase il y a :
– accord sur les paramètres cryptographique,
– échange Diffie-Hellman,
– vérification des identités.
Il est possible de choisir pour la première phase entre deux possibilité d'échange : le *main mode* et *l'aggressive mode* qui diffère par leur rapidité et leur niveau de sécurité.
La seconde phase est en revanche appelée plusieurs fois. En effet les clés qui servent à chiffrer deviennent vulnérables avec le temps ou quand elles sont beaucoup utilisées. Cette phase est donc régulièrement effectuée pour changer certaines clés de sessions.
Il existe en effet trois grands types de clés :
– Les clés de chiffrement de clés : elles sont souvent de très longue durée de vie et peuvent être contenues dans des certificats.
– Les clés maîtresses : elles servent à générer d'autre clés, par exemple une pour l'authentification et une autre pour le chiffrement.
– Les clés de session : ce sont elles qui sont utilisées pour chiffrer, elles ont en général une durée de vie assez faible (de l'ordre de quelques minutes).
Lors de la deuxième phase les échanges sont chiffrés et authentifiés grâce aux éléments négociés lors de la première étape. Les attributs spécifiques à IPsec sont décidés lors de cette phase.

4 Complémentarité entre DNSsec et IPsec

4.1 Présentation

Lors de la mise en place d'une communication IPsec, il y a généralement une négociation IKE. Lors de cette négociation, l'authentification des parties communicantes est nécessaire et peut être réalisée dès lors qu'on possède la clé publique de l'entité avec laquelle on veut initier la communication IPsec. Classiquement ce sont des certificats X.509 [HPFS02] qui sont échangés entre les entités, certificats permettant de faire cette vérification. La vérification du certificat lui-même est nécessaire et fait appel à une infrastructure à clé publique (PKI). L'obtention et la vérification de ces

certificats peuvent nécessiter des mécanismes relativement complexes. Une implémentation de IKE, Racoon, récupère les certificats via le DNS (CERT RR) [Eas99b] pour obtenir les clés publiques.
Une alternative aux certificats est de stocker ces clés nécessaires à IPsec dans un enregistrement spécifique du DNSsec. Cela présente l'avantage de pouvoir vérifier les clés par la chaîne de confiance DNSsec. Cette solution n'est plus possible depuis le RFC 3445 [MR02] précisant que l'enregistrement KEY ne doit contenir que des clés DNSsec. Néanmoins, la création d'un enregistrement IPSECKEY [Ric03] semble se profiler. L'utilisation des enregistrements DNSsec permet de faciliter et de simplifier la phase d'authentification grâce à la disponibilité des clés publiques et à leur intégrité vérifiable par la chaîne de confiance DNSsec, tout en conservant les avantages fournis par les certificats.

4.2 Interaction entre DNSsec et IPsec

Nous proposons un mécanisme similaire à celui qu'utilise l'*Opportunistic Encryption* avec le DNS. L'*Opportunistic Encryption* établit des communications IPsec entre deux entités n'ayant au préalable aucune connaissance spécifique l'une de l'autre. Grâce au DNS, les machines obtiennent les renseignements nécessaires à la mise en place d'une telle communication sécurisée, à savoir l'adresse de la passerelle de sécurité à contacter et la clé publique de cette passerelle. La passerelle de sécurité peut être la machine avec laquelle on veut communiquer. Ces informations sont stockées dans des enregistrements TXT associés à la machine avec laquelle on veut communiquer. Ce mécanisme s'appuyant sur le DNS, il n'y a aucune garantie sur l'intégrité des données reçues ni aucune authentification effectuée. Les clés publiques peuvent aussi être récupérées dans des enregistrements CERT [Eas99b]. Les enregistrements CERT contiennent des certificats qu'il faut ensuite vérifier. Nous proposons d'utiliser DNSsec pour stocker les clés, ainsi que la chaîne de confiance pour garantir intégrité et authentification.

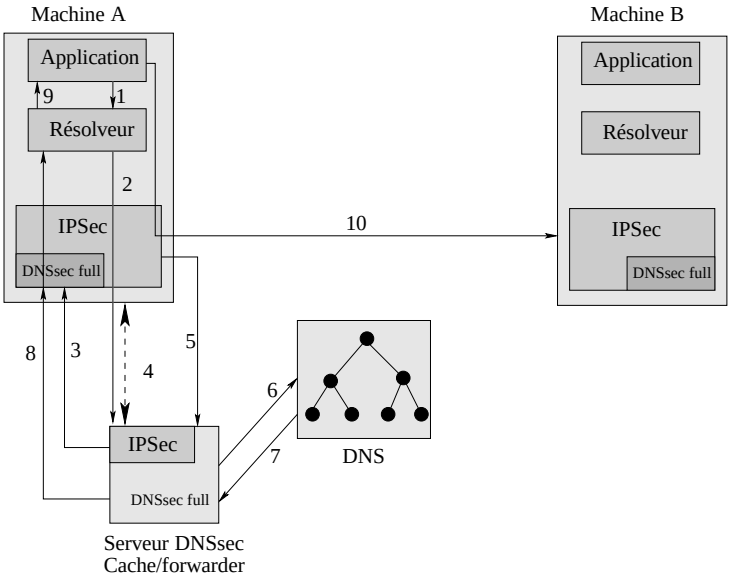


FIG. 4 – *Interaction DNSsec-IPsec lors de l'établissement d'une connexion.*

La figure 4 montre une architecture composée de deux machines, la machine A voulant communiquer avec la machine B, ainsi que le serveur DNSsec cache/forwarder de la zone de la machine A. Un serveur cache/forwarder est chargé de faire suivre les requêtes qu'il reçoit et de garder en cache les réponses. Il ne fait autorité sur aucune zone. Ce type de serveur est utilisé pour améliorer les temps de résolution des requêtes.
Afin d'initier la connexion, la machine A interroge son serveur DNSsec cache/forwarder afin d'obtenir l'adresse IP de la machine B. Pour cela elle fait appel à son résolveur (étape 1). Celui-ci envoie une requête DNS afin d'obtenir l'adresse IP associée au nom de la machine avec laquelle elle veut communiquer (étape 2).
On suppose dans cette architecture que le service en charge d'IPsec possède aussi un module *DNSsec full*. On entend par *DNSsec full* un module capable de demander des enregistrements DNS et d'effectuer une vérification DNSsec complète, c'est-à-dire de valider les enregistrements reçus grâce à leur signature. On suppose aussi que les serveurs DNSsec disposent d'une clé publique présente dans le fichier de zone. Lorsque le serveur cache/forwarder de la zone de la machine A est interrogé, le module IPsec de la machine A récupère la clé publique associée à ce serveur et

la vérifie grâce aux signatures numériques et à la chaîne de confiance DNSsec (étape 3). Si toutes les étapes et tous les enregistrements sont validés, il y a alors établissement d'un canal de communication sécurisé par IPsec entre la machine A et son cache/forwarder (étape 4).
Maintenant, la requête portant sur l'adresse IP de la machine B peut être transmise sur ce canal sécurisé ainsi que la requête portant sur le nom de la passerelle de sécurité et sur la clé publique associées à la machine B (étape 5), le cache/forwarder peut résoudre les requêtes (étape 6 et 7), vérifier les enregistrements reçus en réponse et envoyer les réponses au module IPsec et au résolveur (étape 8) sur le canal sécurisé établi à l'étape 4. Enfin le résolveur délivre la réponse à l'application (étape 9). L'application ayant obtenu l'adresse IP de la machine B ouvre une connexion, le module IPsec se charge de l'établissement du canal sécurisé (étape 10).

4.3 IPsec pour DNSsec

Nous venons de voir qu'il existe bien une complémentarité entre les deux protocoles sécurisés DNSsec et IPsec. Grâce aux interactions possibles entre ces protocoles nous pouvons mettre en place des canaux sécurisés entre une machine et son serveur de noms ou entre deux machines. Il est aussi possible d'utiliser un canal sécurisé pour deux actions spécifiques au DNS : le transfert de zone et les messages de mise à jour dynamique [Wel00] que nous allons étudier maintenant.
Lors de changements dans le fichier de zone (ajout d'un nom, d'une adresse, d'une clé, etc.), les serveurs secondaires doivent mettre à jour leur copie du fichier de zone. Cette opération s'appelle un transfert de zone. Les transferts de zones sont les opérations les plus délicates du DNS, il faut donc pouvoir garantir une sécurité suffisante à de tels échanges.
Afin de protéger les messages de transfert de zone, ceux-ci peuvent être signés par des mécanismes comme TSIG ou SIG(0). Comme nous l'avons vu ces mécanismes nécessitent soit une clé privée partagée entre un serveur de noms primaire et un serveur de noms secondaire (TSIG), soit une paire de clés publique/privée avec la clé privée à disposition (SIG(0)). L'utilisation d'un canal IPsec entre les serveurs secondaires et primaire permet de simplifier le processus en n'utilisant plus ces mécanismes de signatures supplémentaires.
Les messages de mise à jour dynamique concernent les entités capables de modifier des informations du fichier de zone. Il s'agit par exemple d'un serveur DHCP [Dro97] qui peut ajouter ou supprimer des adresses IP. Un canal IPsec peut être mis en place afin d'authentifier une entité et l'autoriser à effectuer ces opérations de mise à jour et afin de protéger les données véhiculées.
Notre méthode consistant en l'établissement d'un canal IPsec entre une machine et son serveur de noms nécessite la présence d'un module *DNSsec full* dans le module IPsec pour l'établissement du canal sécurisé. Même si cela est plus coûteux qu'une connexion simple, le gain en sécurité est important puisqu'il permet une liaison sécurisée pour des machines dont le résolveur n'est pas capable d'effectuer les vérifications de signatures DNSsec. En effet, si le résolveur situé sur la machine est un résolveur basique, c'est-à-dire qu'il ne possède pas le matériel cryptographique et/ou les ressources nécessaires à une vérification DNSsec, ce résolveur doit alors utiliser un drapeau de l'en-tête du message DNS, le bit AD (*Authenticated Data*) [Eas99a]. Ce drapeau spécifie que le serveur de noms interrogé doit répondre uniquement avec des données qu'il a vérifiées et validées et sur un canal sécurisé. Si ces conditions ne sont pas remplies alors les réponses DNS sont rejetées.

4.4 Avantages de notre méthode

Notre méthode présente plusieurs avantages. Tout d'abord elle permet d'utiliser sur les machines des résolveurs plus léger, c'est-à-dire n'intégrant pas toutes les routines pour la vérification cryptographique des signatures. Les résolveurs doivent uniquement comprendre et savoir vérifier la présence du bit AD dans l'en-tête des messages DNS. Il y a ainsi un gain de vitesse à ce niveau car le traitement des requêtes/réponses DNS n'est plus retardé par la vérification cryptographique qui est coûteuse et relativement lente.
De plus, les résolveurs deviennent indépendant des algorithmes de chiffrement utilisés dans le DNSsec. Comme ils n'effectuent pas les vérifications, ils n'ont pas à être mis à jour lorsque les algorithmes utilisés pour chiffrer les données DNS changent. Ces changements ou mise à jour des algorithmes sont alors uniquement fait sur les serveurs de noms.
Le niveau de sécurité des échanges est identique que l'on utilise des certificats ou la méthode présentée ci-dessus avec DNSsec. Le gain se situe au niveau de la gestion qui est simplifiée car il n'y a pas de liste de révocation comme pour les certificats et si une clé est en place dans le DNS, on peut la vérifier grâce à la chaîne de confiance. En ce qui concerne les certificats, il suffit d'un certificat manquant ou invalide pour faire échouer la vérification de la clé et empêcher la négociation de la communication IPsec.

Un autre avantage est la relative simplicité du procédé. En effet, du point de vue des utilisateurs, tout est absolument transparent, tout est géré automatiquement lors de la demande de connexion. Il y a un minimum de configuration, il s'agit juste de placer les clés publiques et les enregistrements nécessaires sur le serveur de noms DNSsec.

5 Conclusions

Après avoir présenté l'extension sécurisée de DNS : DNSsec, ainsi que le protocole IP sécurisé : IPsec, nous avons montré une certaine complémentarité possible dans l'utilisation conjointe de ces deux protocoles. Le modèle exhibé montre qu'il est possible de les faire interagir afin d'obtenir une sécurité optimale des échanges et que cela reste transparent pour les machines communicantes.
Dans notre modèle, IPsec est utilisé pour créer un canal sécurisé nécessaire à certains échanges DNSsec ou pour la sécurisation d'un transfert de zone. Inversement, puisque nous disposons d'une structure sécurisée, DNSsec est utilisé pour mettre en place une communication IPsec.
L'utilisation conjointe de ces deux protocoles amène un gain en terme de sécurité sans alourdir la gestion des communications et avec une configuration minimale, à savoir l'ajout d'enregistrements dans le fichier de zone DNSsec. De plus, il n'est plus nécessaire de faire appel à des certificats pour récupérer les clés publiques, ni à l'infrastructure sous jacente pour les vérifier. Dès qu'une clé est placée dans le fichier de zone par l'administrateur, elle devient vérifiable par la chaîne de confiance DNSsec.
Il semble donc très intéressant de pouvoir combiner ces deux protocoles au profit de la sécurité.

Références

[AL02] P. Albitz and C. Liu. *DNS and BIND*. O'Reilly & Associates, Inc., Sebastopol, CA., fourth edition, Jan 2002.

[Dro97] R. Droms. Dynamic Host Configuration Protocol. RFC 2131, Mar 1997.

[Eas99a] D. Eastlake. Domain Name System Security Extensions. RFC 2535, Mar 1999.

[Eas99b] D. Eastlake. Storing Certificates in the Domain Name System. RFC 2538, Mar 1999.

[Eas00] D. Eastlake. DNS Request and Transaction Signatures (SIG(0)s). RFC 2931, Sep 2000.

[Gie01] R. Gieben. Chain of Trust. Master's Thesis, NLnet Labs, 2001.

[Gun03] O. Gundmundsson. Delegation Signer Resource Record. Draft IETF, last call, Jun 2003.

[HC98] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). RFC 2409, Nov 1998.

[HPFS02] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280, Apr 2002.

[KA98a] S. Kent and R. Atkinson. IP Authentication Header. RFC 2402, Nov 1998.

[KA98b] S. Kent and R. Atkinson. IP Encapsulating Security Payload (ESP). RFC 2406, Nov 1998.

[Kra96] H. Krawczyk. SKEME: A Versatile Secure Key Exchange Mechanism for the Internet. In *Proceedings of the Internet Society Symposium on Network and Distributed System Security*, pages 114–127, Feb 1996.

[Moc87a] P. Mockapetris. Domain Names - Concept and Facilities. RFC 1034, Nov 1987.

[Moc87b] P. Mockapetris. Domain Names - Implementation and Specification. RFC 1035, Nov 1987.

[MR02] D. Massey and S. Rose. Limiting the Scope of the KEY Resource Record (RR). RFC 3445, Dec 2002.

[MSST98] D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet Security Association and Key Management Protocol (ISAKMP). RFC 2408, Nov 1998.

[Orm98] H. Orman. The OAKLEY Key Determination Protocol. RFC 2412, Nov 1998.

[Pos81] J. Postel. Internet Protocol. RFC 0791, Sep 1981.

[Ric03] M. Richardson. A method for storing IPsec keying material in DNS. Draft IETF, work in progress, Sep 2003.

[Sch96] B. Schneier. *Applied cryptography*. John Wiley & Sons, Inc., New York, N.Y., second edition, 1996.

[TDG98] R. Thayer, N. Doraswamy, and R. Glenn. IP Security Document Roadmap. RFC 2411, Nov 1998.

[VGEW00] P. Vixie, O. Gudmunsson, D. Eastlake, and B. Wellington. Secret Key Transaction Authentication for DNS (TSIG). RFC 2845, May 2000.

[Wel00] B. Wellington. Secure Domain Name System (DNS) Dynamic Update. RFC 3007, Nov 2000.